Large Scale Sub-surface Flow Inversion

Shu Wang[1,2]
University of New Mexico, Albuquerque, NM; Los Alamos National Laboratory, Los Alamos, NM

LA-UR-18-27451

Abstract

Sensitivity analysis plays an important role in searching for constitutive parameters (e.g. permeability) sub-surface flow simulations. It requires to solve a dynamic constrained optimization problem. Traditional forward sensitivity analysis is not favorable as the computational cost increases linearly with the product of numbers of parameters and cost functions. Discrete adjoint sensitivity analysis(SA) is gaining popularity due to its computational efficiency. This algorithm is constructed by alternating forward and backward (adjoint) simulation and the computational cost is independent of number of parameters. This property makes it highly powerful when the parameters space is large.

We develop a time-dependent sub-surface flow simulation tool which is capable of running both forward/adjoint simulations as well as non-linear optimization based on Portable, Extensible toolkit for Scientific Computation (PETSc). We also present a detailed performance of parallelization on distributed/shared memory computing architectures.

Problem Statement

◆ Convection equation for sub-surface flow

$$F(u,p) = \frac{du}{dt} - \nabla\big(p(x)\cdot\nabla u(x)\big) - b(x)$$

- *u(x)* pressure
- *b(x)* sink/source
- *p(x)* permeability/diffusivity (unknown parameters)
- *F(u,p)* residual

◆Optimization problem

$$\min J(u,p)$$
$$s.t.\ F^d(u,p) = 0, \text{ where}$$

$$J(u,p) = \frac{1}{2}\sum_{i=1}^{N}\left\|u_i(t_{T_F}) - u_i^{obs}(t_{T_F})\right\|$$

$$F^d(u,p) = \sum_{k=1}^{T_F}\lambda_k^T[u(t_k) - G(t(t_{k-1}))]$$

- F^d discretized residual
- $J(u,p)$ objective function
- λ_k Lagrange multipliers
- $u_i^{obs}(t_{T_F})$ observation data for last time snapshot
- G integrators of choice
- T_F Total time steps
- N degrees of freedom

◆Discrete adjoint sensitivity analysis

Forward +inversion

Example : backward Euler integrator

Forward

$$u_{n+1} = u_n + dt\cdot F(t_n, u_{n+1})$$

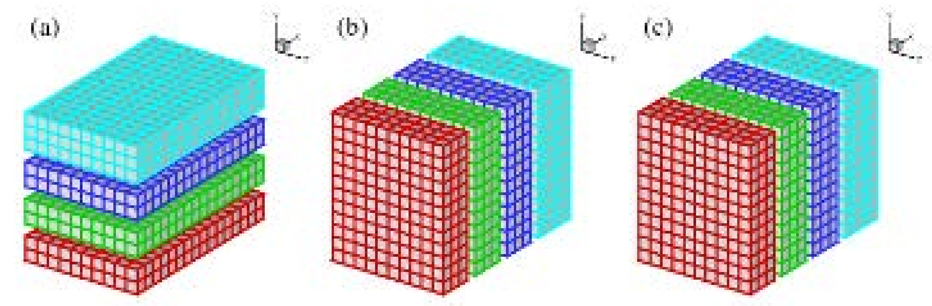
t_0 t_n t_{n+1} t_F

$$\lambda_n = \lambda_{n+1} + dt\cdot F_u(t_{n+1}, u_{n+1})^T\cdot\lambda_n$$

Adjoint

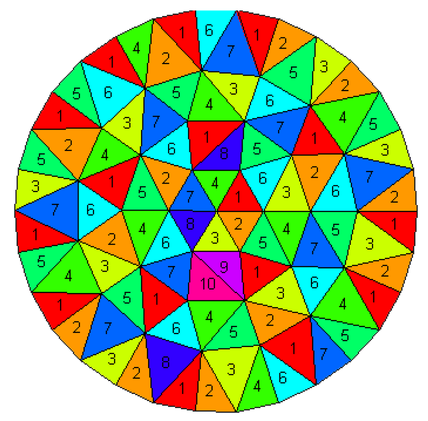
Technical Approach

- Domain decomposition
  - Grid partition (ParMetis)
  - MPI level only
- Thread level parallelism
  - Greedy graph coloring



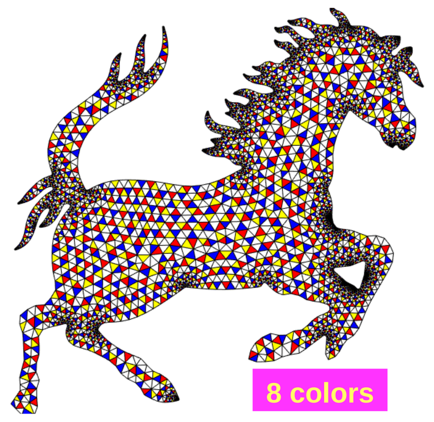
Greedy Coloring Algorithm

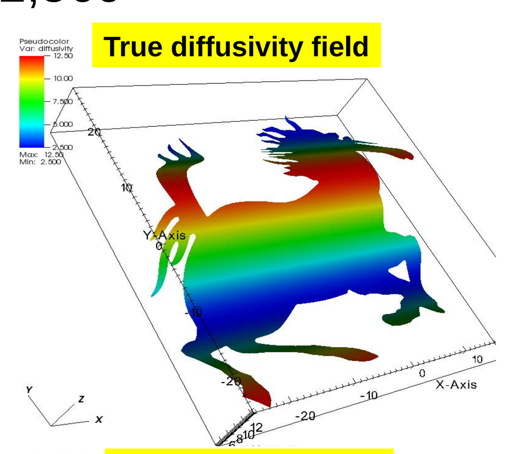
1. Get the next element in the mesh
2. Traverse all neighbors using $L(G^C)$, and see what colors cannot be used
3. Color this element with the next available color
4. If this is not the last element, goto 1

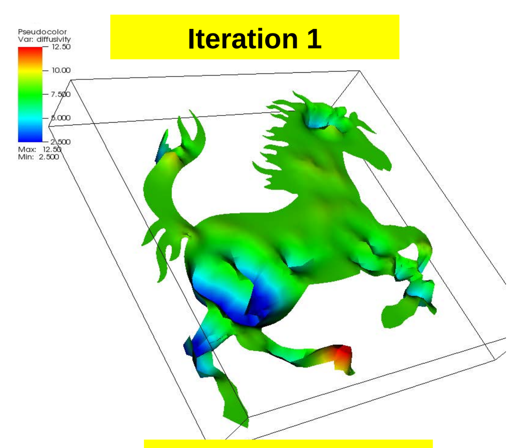


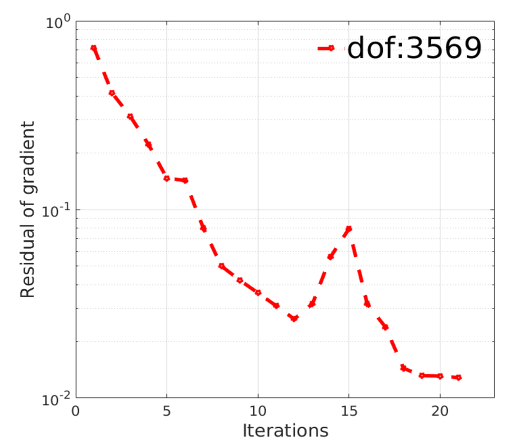
Simulation Results

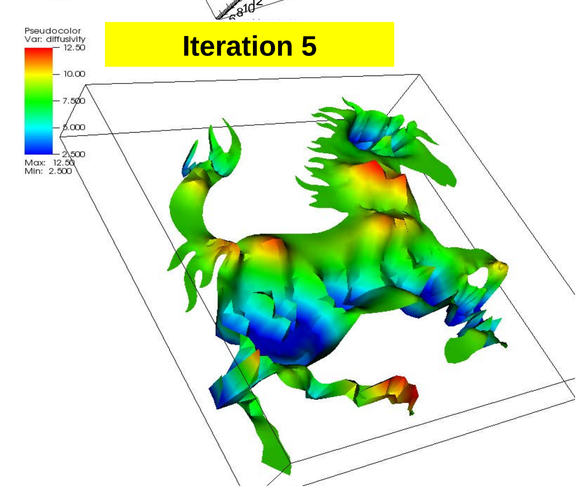
Triangular Mesh, dof 2,569

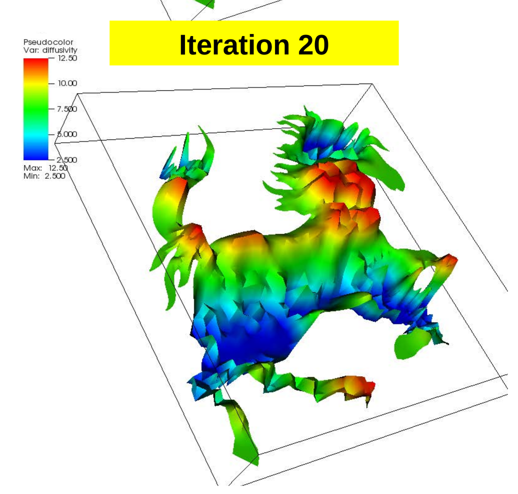




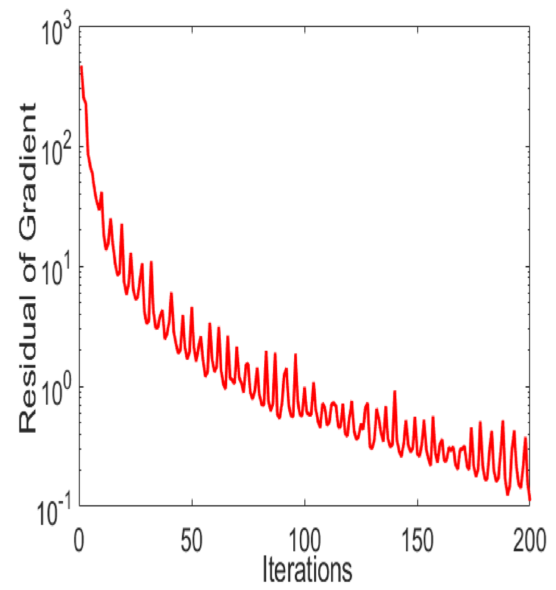


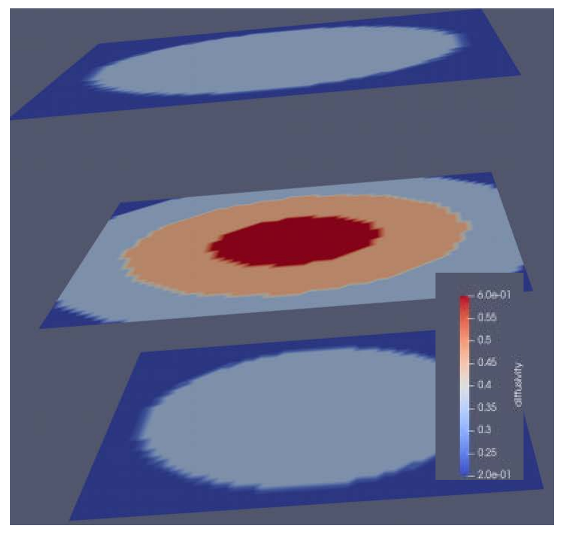


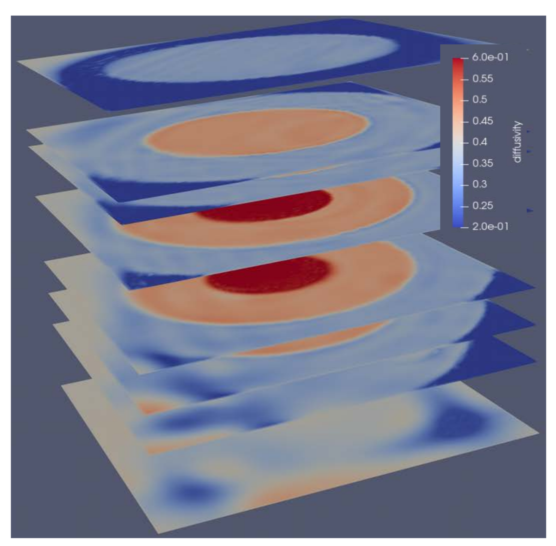




Hexahedron Mesh, dof 1,000,000

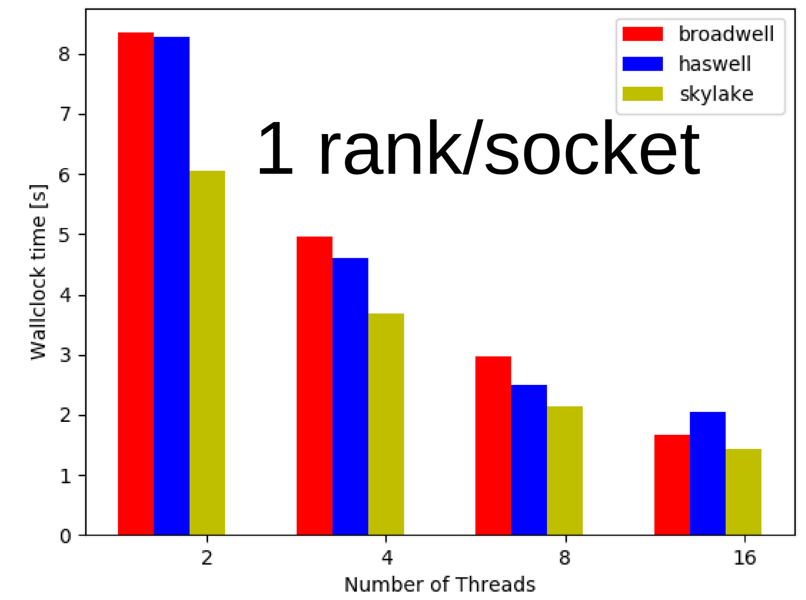




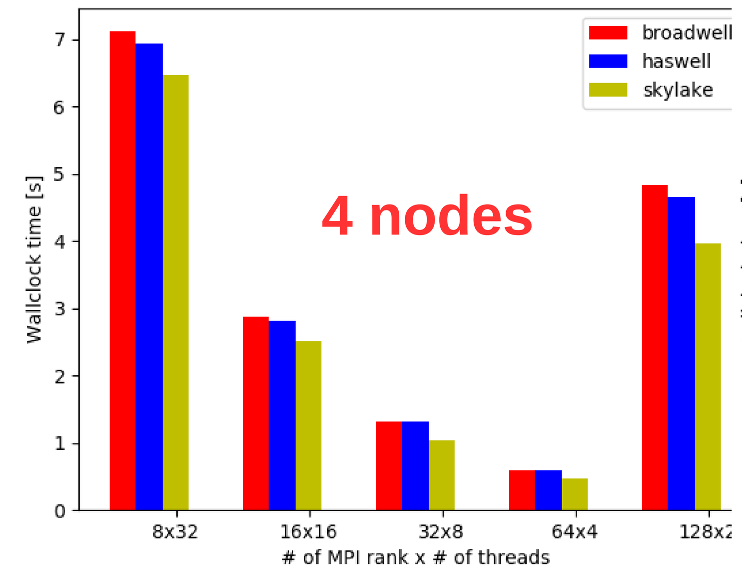


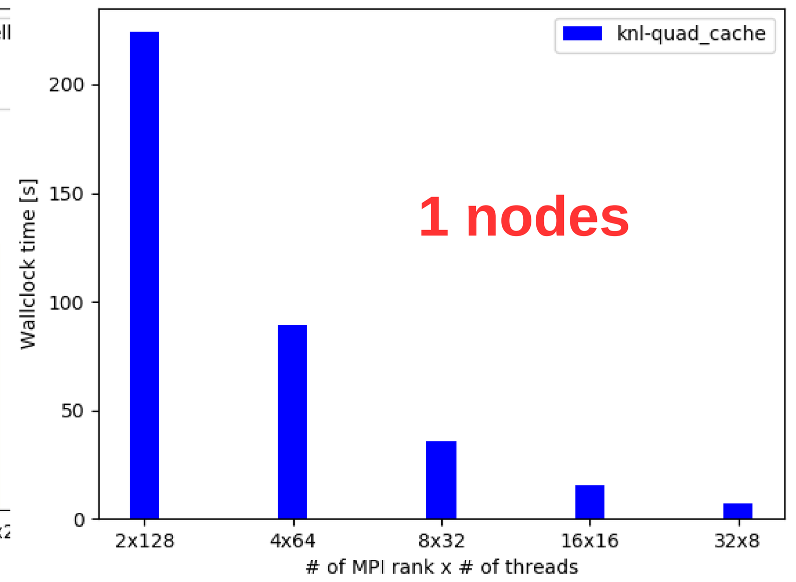
Parallel Performance

- On-node performance

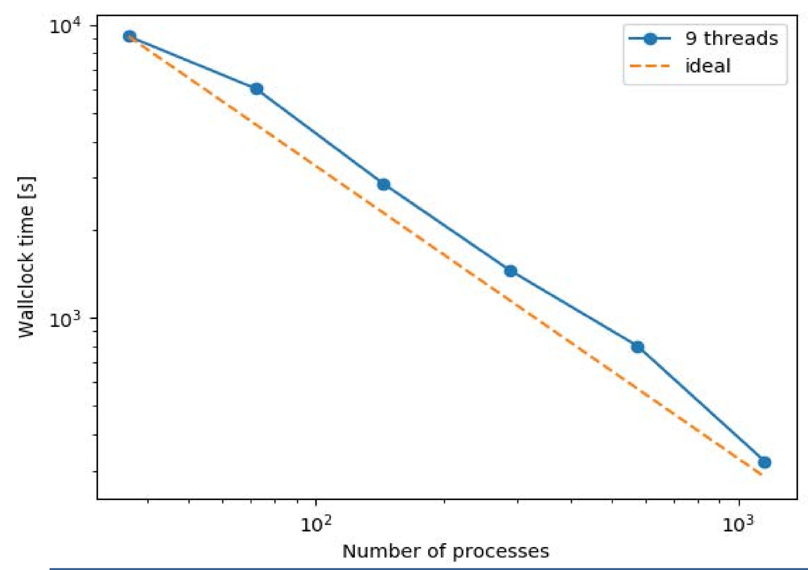


- Distributed Memory / Many-core Performance





- Strong Scaling Performance (MPI+OpenMP)



- Intel Broadwell Node (18 cores/socket, 2 sockets/node)
- Fully utilize the node (e.g. 4 ranks/node + 9 threads/rank)
- Scales well up to 1,152 cores!

Conclusion

1. Implemented a sub-surface flow solver based on discrete adjoint sensitivity analysis
2. Realization of multi-level parallelization by MPI and OpenMP.
3. Excellent scaling performance were observed.

Acknowledgement

1. Many thanks to the parallel Computing Research Internship mentors Satish Karra, Hai Ah Nam, Bob Robey, Kris Garrett, Eunmo Ko, and Luke Van Roekel.
2. This work was performed on Darwin, grizzly systems and supported by LANL under contract